

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-085112

(43)Date of publication of application : 31.03.1995

(51)Int.Cl.

G06F 17/50

(21)Application number : 05-224183

(71)Applicant : HITACHI LTD

(22)Date of filing : 09.09.1993

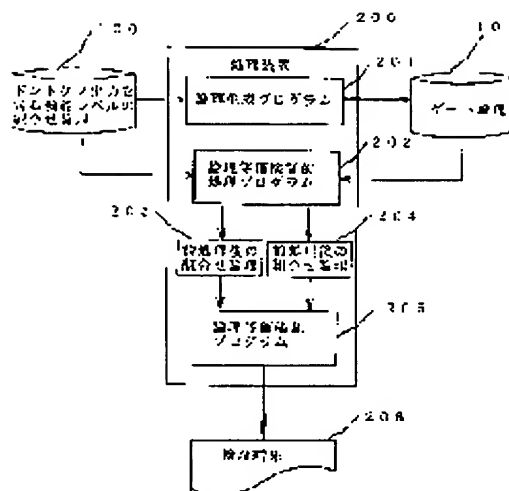
(72)Inventor : MORIKAWA NAOTO  
NIIYA TAKAO  
TANDAI MIYAKO  
MIZOGAMI YOSHITO

## (54) LOGIC EQUIVALENT VERIFICATION METHOD

### (57)Abstract:

**PURPOSE:** To attain the logic equivalent verification for a combination logic of a function level including a don't care output and a gate logic generated automatically from the combination logic by eliminating the logic uncertainty of the logic caused by the don't care output.

**CONSTITUTION:** A logic generation program 201, a logic equivalent verification pre-processing program 202, and a logic equivalent verification program 205 work respectively on a processing unit 200. The program 201 inputs a function logic file 100, generates a gate logic based on a combination logic of a function level including a don't care output and outputs to the gate logic file 101. The program 202 inputs the files 100, 101, executes a logic equivalent verification pre-processing to each of a combination logic of a function level including a don't care output and a gate logic and provides combination logics 203, 204 after the pre-processing to a main storage device of the processing unit 200. The program 205 inputs the combination logics 203, 204 in the main storage device, converts the both into bisection decision graph to decide the match of structures of the graphs and provides the output of a verification result 206.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-85112

(43) 公開日 平成7年(1995)3月31日

(51) Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 17/50

7623-5L

G 0 6 F 15/ 60

3 6 0 D

審査請求 未請求 請求項の数 5 O L (全 18 頁)

(21) 出願番号 特願平5-224183

(22) 出願日 平成5年(1993)9月9日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 森川 直人

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 新舎 隆夫

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 旦代 三弥子

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

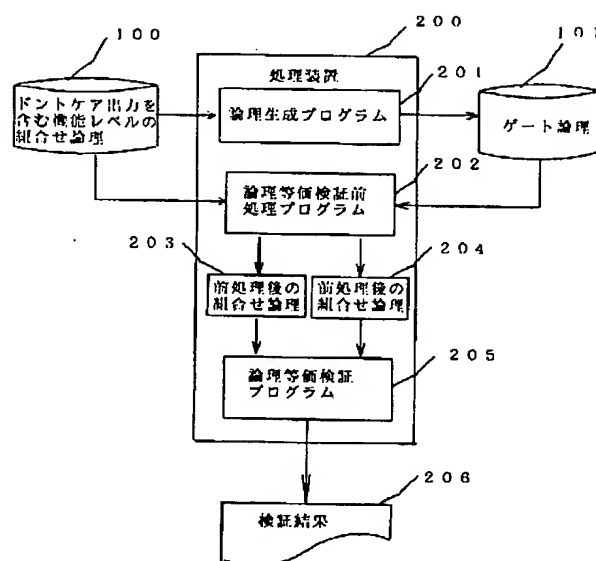
(54) 【発明の名称】 論理等価検証方法

(57) 【要約】

【構成】 処理装置200上で、論理生成プログラム201、論理等価検証前処理プログラム202、論理等価検証プログラム205が各々動作する。プログラム201は、機能論理ファイル100を入力し、ドントケア出力を含む機能レベルの組合せ論理からゲート論理を生成し、ゲート論理ファイル101に出力する。プログラム202は、ファイル100と101を入力し、ドントケア出力を含む機能レベルの組合せ論理とゲート論理の各々に対して、論理等価検証前処理を行い、前処理後の組合せ論理203と204を処理装置200の主記憶に出力する。プログラム205は、主記憶上の組合せ論理203と204を入力し、両者を二分決定グラフに変換してグラフの構造一致判定を行い、検証結果206を出力する。

【効果】 ドントケア出力に起因する論理の不確定性を取り除くことができるので、ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証が可能である。

図2



100: 機能論理ファイル  
101: ゲート論理ファイル

1

## 【特許請求の範囲】

【請求項1】 ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を行うシステムにおいて、該組合せ論理と該ゲート論理を入力する第一ステップと、該組合せ論理の確定出力の論理部分をブール式表現化してブール式セットAを作成する第二ステップと、該組合せ論理のドントケア出力の論理部分をブール式表現化してブール式セットBを作成する第三ステップと、該ゲート論理をブール式表現化してブール式セットYを作成する第四ステップと、出力変数単位に該ブール式セットAと該ブール式セットBの論理和演算を行ってブール式セットDを作成する第五ステップと、出力変数単位に該ブール式セットYと該ブール式セットBの論理和演算を行ってブール式セットEを作成する第六ステップと、該ブール式セットDと該ブール式セットEの論理等価検証を行う第七ステップを含むことを特徴とする論理等価検証方法。

【請求項2】 ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を行うシステムにおいて、該組合せ論理と該ゲート論理を入力する第一ステップと、該組合せ論理の確定出力の論理部分をブール式表現化してブール式セットAを作成する第二ステップと、該組合せ論理のドントケア出力の論理部分をブール式表現化してブール式セットBを作成する第三ステップと、出力変数単位に該ブール式セットBからフィルターセットFを作成する（該ブール式セットBがn入力m出力のとき、出力変数単位にm個のn入力n出力の組合せ論理（フィルター）のセットFを作成する）第五ステップと、該ブール式セットAと該フィルターセットFを接続して（該ブール式セットAがn入力m出力のとき、出力変数単位に出力変数Yj（j=1, ..., m）を表すブール式の入力変数Xi（i=1, ..., n）を該出力変数Yjに対応するフィルターの出力変数Zi（i=1, ..., n）で置き換えて）ブール式セットGを作成する第六ステップと、該ブール式セットYと該フィルターセットFを接続してブール式セットHを作成する第七ステップと、該ブール式セットGと該ブール式セットHの論理等価検証を行う第八ステップを含むことを特徴とする論理等価検証方法。

【請求項3】 ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を行うシステムにおいて、該組合せ論理と該ゲート論理を入力する第一ステップと、該組合せ論理の確定出力の論理部分をブール式表現化してブール式セットAを作成する第二ステップと、該組合せ論理のドントケア出力の論理部分をブール式表現化してブール式セットBを作成する第三ステップと、出力変数単位に該ブール式セットAと該ブール式セットYのXNOR(Exclusive NOR)またはEOR(Exclusive OR)論理演算を行ってブール式セットIを作成する第五ステップと、出力変数単位に該

2

ブール式セットIと該ブール式セットBの論理和演算を行ってブール式セットJを作成する第六ステップと、該ブール式セットJの各出力変数が恒等的に1である（EOR論理演算を行った場合は0である）ことを判定する第七ステップを含むことを特徴とする論理等価検証方法。

【請求項4】 ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を行うシステムにおいて、該組合せ論理と該ゲート論理を入力する第一ステップと、該組合せ論理の確定出力の論理部分をブール式表現化してブール式セットAを作成する第二ステップと、該組合せ論理のドントケア出力の論理部分をブール式表現化してブール式セットBを作成する第三ステップと、出力変数単位に該ブール式セットBからフィルターセットFを作成する（該ブール式セットBがn入力m出力のとき、出力変数単位にm個のn入力n出力の組合せ論理（フィルター）のセットFを作成する）第五ステップと、出力変数単位に該ブール式セットAと該ブール式セットYのXNOR(Exclusive NOR)またはEOR(Exclusive OR)論理演算を行ってブール式セットIを作成する第六ステップと、該ブール式セットIと該フィルターセットFを接続してブール式セットKを作成する第七ステップと、該ブール式セットKの各出力変数が恒等的に1である（EOR論理演算を行った場合は0である）ことを判定する第八ステップを含むことを特徴とする論理等価検証方法。

【請求項5】 請求項2に記載の論理等価検証方法において、上記第五ステップは、n入力m出力の該ブール式セットBの出力変数Yj（j=1, ..., m）を表すブール式Bjからn入力n出力の出力変数Zi（i=1, ..., n）の組合せ論理（フィルター）を作成するとき、各出力変数Ziの初期値を該ブール式セットBの入力変数Xiとし、該ブール式Bjのコンプリメント' Bjを算出し、該ブール式Bjの各積項について、該積項を積項t0として選択し、該コンプリメント' Bjの一つの積項を積項t1として選択し、各入力変数Xiについて、該積項t0とt1が該入力変数Xiを共有しないで、該積項t1がリテラルXiを含んでいるならば、該出力変数Ziを表すブール式において、リテラルXiをXi+(t0/' Xi)で置き換え、該積項t0とt1が該入力変数Xiを共有しないで、該積項t1がリテラルXiを含んでいないならば、該出力変数Ziを表すブール式に' (t0/Xi)を乗算する過程を含むことを特徴とする論理等価検証方法。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、論理等価検証方法に係り、特に組合せ論理の論理等価検証方法に関する。

【0002】

【従来の技術】 従来の組合せ論理の論理等価検証方法は、大きく三つに分けられる、第一の方法は、全ての入力パターンをシミュレーションして検証する方法であ

3

る。第二の方法は、二つの組合せ論理の各出力をXNOR (Exclusive NOR) で接続し、正当化により各出力が0になる入力パターンが存在するか否かを調べる方法であり、このような方法の一つが、IEEE ITC-86, pp. 350-359(1986)に記載されている。第三の方法は、二つの組合せ論理を2分決定グラフ(BDD)に変換してグラフの構造一致判定を行う方法であり、このような方法の一つが、IEEE TRANSACTION ON COMPUTERS, VOL. C-35, NO. 8, AUG. pp. 677-691(1986)に記載されている。

【0003】

【発明が解決しようとする課題】ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理(ドントケア出力は、論理生成システムにより、0出力または1出力に最適化されている)の論理等価検証を行う場合、上記第一の方法は、入力パターン数が膨大になり、論理シミュレーションが実際には実施不可能という問題があった。また、上記第二と第三の方法は、ドントケア出力が考慮されていないので、ドントケア出力に起因して、検証結果が論理不一致になる場合があるという問題があった。

【0004】本発明の目的は、上記第二と第三の方法に、ドントケア出力に起因する論理の不確定性を取り除くための論理等価検証前処理を追加することにより、ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を可能にすることにある。

【0005】

【課題を解決するための手段】本発明の第1の特徴は、ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理を入力する第一ステップと、該組合せ論理の確定出力の論理部分をブール式表現化してブール式セットAを作成する第二ステップと、該組合せ論理のドントケア出力の論理部分をブール式表現化してブール式セットBを作成する第三ステップと、該ゲート論理をブール式表現化してブール式セットYを作成する第四ステップと、出力変数単位に該ブール式セットAと該ブール式セットBの論理和演算を行ってブール式セットDを作成する第五ステップと、出力変数単位に該ブール式セットYと該ブール式セットBの論理和演算を行ってブール式セットEを作成する第六ステップと、該ブール式セットDと該ブール式セットEの論理等価検証を行う第七ステップを含む論理等価検証方法にある。

【0006】また、本発明の第2の特徴は、上述の第一～第四ステップと、出力変数単位に該ブール式セットBからフィルターセットFを作成する(該ブール式セットBがn入力m出力のとき、出力変数単位にm個のn入力n出力の組合せ論理(フィルター)のセットFを作成する)第五ステップと、該ブール式セットAと該フィルターセットFを接続して(該ブール式セットAがn入力m

4

出力のとき、出力変数単位に出力変数 $Y_j$ ( $j=1, \dots, m$ )を表すブール式の入力変数 $X_i$ ( $i=1, \dots, n$ )を該出力変数 $Y_j$ に対応するフィルターの出力変数 $Z_i$ ( $i=1, \dots, n$ )で置き換えて)ブール式セットGを作成する第六ステップと、該ブール式セットYと該フィルターセットFを接続してブール式セットHを作成する第七ステップと、該ブール式セットGと該ブール式セットHの論理等価検証を行う第八ステップを含む論理等価検証方法にある。

【0007】また、本発明の第3の特徴は、上述の第一～第四ステップと同一の第一～第四ステップと、出力変数単位に該ブール式セットAと該ブール式セットYのXNOR(Exclusive NOR)(または、EOR(Exclusive OR))論理演算を行ってブール式セットIを作成する第五ステップと、出力変数単位に該ブール式セットIと該ブール式セットBの論理和演算を行ってブール式セットJを作成する第六ステップと、該ブール式セットJの各出力変数が恒等的に1である(EOR論理演算を行った場合は0である)ことを判定する第七ステップからなるようにしたものである(手段3)。

【0008】また、本発明の第4の特徴は、上述の第一～第四ステップと、上記手段の第五ステップと同一の第六ステップと、出力変数単位に該ブール式セットBからフィルターセットFを作る第五ステップと(第2の特徴における第五ステップと同様)、出力変数単位毎に該ブール式セットAと該ブール式セットYのXNORまたはEOR論理演算を行なってブール式セットIを作成する第六ステップ(第3の特徴における第五ステップと同様)、該ブール式セットIと該フィルターセットFを接続してブール式セットKを作成する第七ステップと、該ブール式セットKの各出力変数が恒等的に1である(EOR論理演算を行った場合は0である)ことを判定する第八ステップからなるようにしたものである(手段4)。

【0009】

【作用】上記特徴1～4にしたがう方法はいずれも、ドントケア出力に起因する論理の不確定性を取り除き、従来の論理等価検証方法と併用することにより、ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証を可能にする。

【0010】

【実施例】以下、本発明の実施例を図面により詳細に説明する。実施例は四つあり、第一と第二の実施例は第一の論理等価検証システム構成を前提にし、第三と第四の実施例は第二の論理等価検証システム構成を前提にする。

【0011】図2は、第一の論理等価検証システム構成を表す図である。処理装置200上で、論理生成プログラム201、論理等価検証前処理プログラム202、論理等価検証プログラム205が各々動作する。プログラム201は、機能論理ファイル100を入力し、ドント

ケア出力を含む機能レベルの組合せ論理からゲート論理を生成し、ゲート論理ファイル101に出力する。プログラム202は、ファイル100と101を入力し、ドントケア出力を含む機能レベルの組合せ論理とゲート論理の各々に対して、論理等価検証前処理を行い、前処理後の組合せ論理203と204を処理装置200の主記憶に出力する。プログラム205は、主記憶上の組合せ論理203と204を入力し、両者を二分決定グラフに変換してグラフの構造一致判定を行い、検証結果206を出力する。上記において、プログラム202が本発明に関する部分である。

【0012】図1は、本発明に基づく第一の実施例を表す論理等価検証前処理のフローチャートである。この図に基づき、論理等価検証前処理の処理手順を順次説明する。

【0013】ステップ102：本ステップは、機能論理ファイル100とゲート論理ファイル101から、ドントケア出力を含む機能レベルの組合せ論理とゲート論理を入力する。

【0014】図3は、ドントケア出力を含む機能レベルの組合せ論理（真理値表）の例を表し、図4は、真理値表300から生成されたゲート論理の例を表す。

【0015】ステップ103：本ステップは、ステップ102で入力した機能レベルの組合せ論理において、確定出力の組合せ論理部分をブール式表現化し、ブール式セットAを作成する。

【0016】図5は、真理値表300から得られたブール式セットAを表す。

【0017】ステップ104：本ステップは、ステップ102で入力した機能レベルの組合せ論理において、ドントケア出力の組合せ論理部分をブール式表現化し、ブール式セットBを作成する。

【0018】図6は、真理値表300から得られたブール式セットBを表す。

【0019】ステップ105：本ステップは、ステップ102で入力したゲート論理をブール式表現化し、ブール式セットYを作成する。

【0020】図7は、ゲート論理400から得られたブール式セットYを表す。

【0021】ステップ106：本ステップは、出力変数単位に、ブール式セットAとブール式セットBの論理和演算を行い、ブール式セットDを作成する。

【0022】図8は、ブール式セットA500とブール式セットB600を論理和演算したブール式セットDを表す。

【0023】ステップ107：本ステップは、出力変数単位に、ブール式セットYとブール式セットBの論理和演算を行い、ブール式セットEを作成する。

【0024】図9は、ブール式セットY700とブール式セットB600を論理和演算したブール式セットEを

表す。

【0025】ステップ108：本ステップは、ブール式セットDとEを論理等価検証前処理後の組合せ論理203と204として主記憶に出力する。

【0026】図10は、本発明に基づく第二の実施例を表す論理等価検証前処理のフローチャートである。図10は、図1のステップ106～108をステップ1000～1003で置き換えたものである。ステップ1000～1003の処理手順を順次説明する。

【0027】ステップ1000：本ステップは、ステップ104で得られたブール式セットBからフィルターセットFを作成する。ここで、フィルターは、ドントケア出力を取り除くためのn入力n出力の組合せ論理であり、ブール式セットBの出力単位に作成される。また、nは入力変数の総数である。

【0028】図11は、フィルター作成処理のフローチャートである。この図に基づき、フィルター作成処理の処理手順を順次説明する。

【0029】ステップ1101：本ステップは、ブール式セットBの出力変数 $Y_j$  ( $j=1, \dots, m$ )のドントケア出力の組合せ論理部分のブール式 $B_{j1100}$ を入力する。ここで、mは出力変数の総数である。

【0030】ステップ1102：本ステップは、出力変数 $Y_j$ のフィルターのブール式セット $Z_{i1115}$ の初期値をブール式セットBの入力変数 $X_i$ とする ( $i=1, \dots, n$ )。ここで、nは出力変数の総数である。

【0031】ステップ1103：本ステップは、ブール式 $B_{j1100}$ のコンプリメント $\neg B_j$ を算出する。

【0032】ステップ1104：本ステップは、ブール式 $B_{j1100}$ の未選択の一つの積項を $t_0$ として選択する。ここで、 $t_0$ は、ブール式 $B_{j1100}$ の最も左側にある積項から順に選択する。

【0033】ステップ1105：本ステップは、コンプリメント $\neg B_j$ の一つの積項を $t_1$ として選択する。ここで、 $t_1$ は、リテラル数が最少の積項を、この候補が複数ある場合は、積項 $t_0$ と最も多くのリテラルを共有する積項を、この候補が複数ある場合は、コンプリメント $\neg B_j$ の最も左側にある積項を選択する。

【0034】ステップ1106：本ステップは、作業用変数 $l$ の初期値を1にする。

【0035】ステップ1107：本ステップは、積項 $t_0$ と $t_1$ が入力変数 $X_i$ を「共有」しているか否かを判定し、「共有」していれば、ステップ1111に進み、そうでなければステップ1108に進む。図30に、積項 $t_0$ と $t_1$ における入力変数 $X_i$ の現われ方の組合せ全9通りの各々に対し、変数 $X_i$ を「共有」しているか否かを示す。

【0036】図30は、積項 $t_0$ と $t_1$ の入力変数 $X_i$ に関する「共有」関係の定義を表す。

【0037】ステップ1108：本ステップは、積項 $t_1$

7

がリテラル $X_i$ を含んでいるか否かを判定し、含んでいれば、ステップ1109に進み、そうでなければ、ステップ1110に進む。

【0038】ステップ1109：本ステップは、フィルターの第 $i$ 番目の出力変数 $Z_i$ を表すブール式において、リテラル $X_i$ を $X_i + (t0/X_i)$ で置き換え、ステップ1111に進む。ここで、 $t0/X_i$ は、積項 $t0$ にリテラル $X_i$ が含まれていれば、 $t0$ を $X_i$ で代数的に割って得られる積項を、そうでなければ $t0$ を表す。 $(t0/X_i)$ についても同様。）

ステップ1110：本ステップは、フィルターの第 $i$ 番目の出力変数 $Z_i$ を表すブール式に $(t0/X_i)$ を乗算したブール式を $Z_i$ とする。

【0039】ステップ1111：本ステップは、作業用変数 $i$ の値に1を加算する。

【0040】ステップ1112：本ステップは、作業用変数 $i$ の現在の値と $n$ （入力変数の総数）を比較し、 $i > n$ であれば、ステップ1113に進み、そうでなければ、ステップ1107に進む。

【0041】ステップ1113：本ステップは、ブール式 $B_{j1100}$ の積項が全て $t0$ として処理されたか否かを判定し、処理済みであれば、ステップ1114に進み、そうでなければ、ステップ1104に進む。

【0042】ステップ1114：本ステップは、フィルターのブール式セット $Z_i (i=1, \dots, n)$ 1115を出力する。

【0043】上記のフィルター作成処理を具体例を用いて説明する。ブール式 $B_{j1100}$ が出力変数 $Y1$ のブール式601（図6）の場合のフィルター作成処理は以下の通りである。ステップ1101により、ブール式601を入力する。ステップ1102により、処理結果1400（図14）を得る。ステップ1103により、ブール式601のコンプリメント1501（図15）を得る。ステップ1104と1105（一回目）により、処理結果1600（図16）を得る。ステップ1106により、 $i=1$ とする。ステップ1107において、処理結果1600の $t0$ と $t1$ は入力変数 $X1$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=2$ となる。ステップ1112において、 $i=2 < n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果1600の $t0$ と $t1$ は入力変数 $X2$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=3$ となる。ステップ1112において、 $i=n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果1600の $t0$ と $t1$ は入力変数 $X3$ を共有していないので、ステップ1108に進む。ステップ1108において、処理結果1600の $t1$ はリテラル $X3$ を含んでいるので、ステップ1109に進む。ステップ1109により、フィルターの第三番目の出力変数 $Z3$ を表すブール式において、リテラル $X3$ を $X3 + ((t0/X3) = X3 + X1' X2)$ に置き換える。ステップ1111により、 $i=4$ となる。ステップ1112において、 $i=4 > n=3$ であるので、ステップ1113に進む。ステップ1113において、ブール式601の積項は全て $t0$ として処理されていないので、ステップ1104に進む。上記の一連のステップにより、出力変数 $Y1$ のフィルターの二回目の中間結果1900（図19）を得る。ステップ1104と1105（三回目）により、処理結果2000（図20）を得る。ステップ1106により、 $i=1$ とする。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X1$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=2$ となる。ステップ1112において、 $i=2 < n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X2$ を共有していないので、ステップ1108に進む。ステップ1108において、処理結果2000の $t1$ はリテラル $X2$ を含んでいないので、ステップ1110に進む。ステップ1110において、 $(t0/X2) = (X1X3) = X1 + X3$ であるので、 $Z1 = (X1 + X3) X2$ を得る。ステップ1111により、 $i=3$ となる。ステップ1112において、 $i=n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X3$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=4$ と

8

式において、リテラル $X3$ を $X3 + ((t0/X3) = X3 + X1' X2)$ に置き換える。ステップ1111により、 $i=4$ となる。ステップ1112において、 $i=4 > n=3$ であるので、ステップ1113に進む。ステップ1113において、ブール式601の積項は全て $t0$ として処理されていないので、ステップ1104に進む。上記の一連のステップにより、出力変数 $Y1$ のフィルターの一回目の中間結果1700（図17）を得る。ステップ1104と1105（二回目）により、処理結果1800（図18）を得る。ステップ1106により、 $i=1$ とする。ステップ1107において、処理結果1800の $t0$ と $t1$ は入力変数 $X1$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=2$ となる。ステップ1112において、 $i=2 < n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果1800の $t0$ と $t1$ は入力変数 $X2$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=3$ となる。ステップ1112において、 $i=n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果1800の $t0$ と $t1$ は入力変数 $X3$ を共有していないので、ステップ1108に進む。ステップ1108において、処理結果1800の $t1$ はリテラル $X3$ を含んでいるので、ステップ1109に進む。ステップ1109により、フィルターの第三番目の出力変数 $Z3$ を表すブール式において、リテラル $X3$ を $X3 + ((t0/X3) = X3 + X1' X2)$ に置き換える。ステップ1111により、 $i=4$ となる。ステップ1112において、 $i=4 > n=3$ であるので、ステップ1113に進む。ステップ1113において、ブール式601の積項は全て $t0$ として処理されていないので、ステップ1104に進む。上記の一連のステップにより、出力変数 $Y1$ のフィルターの二回目の中間結果1900（図19）を得る。ステップ1104と1105（三回目）により、処理結果2000（図20）を得る。ステップ1106により、 $i=1$ とする。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X1$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=2$ となる。ステップ1112において、 $i=2 < n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X2$ を共有していないので、ステップ1108に進む。ステップ1108において、処理結果2000の $t1$ はリテラル $X2$ を含んでいないので、ステップ1110に進む。ステップ1110において、 $(t0/X2) = (X1X3) = X1 + X3$ であるので、 $Z1 = (X1 + X3) X2$ を得る。ステップ1111により、 $i=3$ となる。ステップ1112において、 $i=n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2000の $t0$ と $t1$ は入力変数 $X3$ を共有しているので、ステップ1111に進む。ステップ1111により、 $i=4$ と

9

なる。ステップ1112において、 $i=4>n=3$ であるので、ステップ1113に進む。ステップ1113において、ブール式601の積項は全て10として処理されたので、本処理は終了する。上記の一連のステップにより、出力変数Y1のフィルター1200（図12）を得る。

【0044】また、ブール式B11100が出力変数Y2のブール式602（図6）の場合のフィルター作成処理は以下の通りである。ステップ1101により、ブール式602を入力する。ステップ1102により、処理結果1400（図14）を得る。ステップ1103により、ブール式602のコンプリメント1502（図15）を得る。ステップ1104と1105により、処理結果2100（図21）を得る。ステップ1106により、 $i=1$ とする。ステップ1107において、処理結果2100の10と11は入力変数X1を共有していないので、ステップ1108に進む。ステップ1108において、処理結果2100の11はリテラルX1を含んでいないので、ステップ1110に進む。ステップ1110により、 $(10/X1) = (X2 \cdot X3) = X2 + X3$ であるので、 $Z1 = (X2 + X3) \cdot X1$ を得る。ステップ1111により、 $i=2$ となる。ステップ1112において、 $i=2<n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2100の10と11は入力変数X2を共有しているので、ステップ1111に進む。ステップ1111により、 $i=3$ となる。ステップ1112において、 $i=n=3$ であるので、ステップ1107に進む。ステップ1107において、処理結果2100の10と11は入力変数X3を共有しているので、ステップ1111に進む。ステップ1111により、 $i=4$ となる。ステップ1112において、 $i=4>n=3$ であるので、ステップ1113に進む。ステップ1113において、ブール式602の積項は全て10として処理されたので、本処理は終了する。上記の一連のステップにより、出力変数Y2のフィルター1300（図13）を得る。

【0045】ステップ1001：本ステップは、ブール式セットAとフィルターセットFを接続し、ブール式セットGを作成する。ここで、ブール式セットAとフィルターセットFの接続は、ブール式セットAの出力変数単位に、当該出力変数を表すブール式の入力変数Xiを、当該出力変数に対応するフィルターの出力変数Ziで置き換えることである( $i=1, \dots, n$ )。

【0046】図22は、ブール式セットA500（ブール式501と502）とフィルターセットF（フィルター1200と1300）を接続したブール式セットGを表す。

【0047】ステップ1002：本ステップは、ブール式セットYとフィルターセットFを接続し、ブール式セットHを作成する。

【0048】図22は、ブール式セットY700（ブー

10

ル式701と702）とフィルターセットF（フィルター1200と1300）を接続したブール式セットHを表す。

【0049】ステップ1003：本ステップは、ブール式セットGとHを論理等価検証前処理後の組合せ論理203と204として主記憶に出力する。

【0050】図25は、第二の論理等価検証システム構成を表す図である。処理装置200上で、論理生成プログラム201と論理等価検証プログラム2500が各々動作する。プログラム201は、機能論理ファイル100を入力し、ドントケア出力を含む機能レベルの組合せ論理からゲート論理を生成し、ゲート論理ファイル101に出力する。プログラム2500は、論理等価検証前処理2501と論理等価検証処理2503からなる。処理2501は、ファイル100と101を入力し、ドントケア出力を含む機能レベルの組合せ論理とゲート論理から論理等価検証モデル2502を作成する。処理2503は、論理等価検証モデル2502を評価し（正当化により当該モデルの各出力が0になるパターンが存在するか否かを調べ）、検証結果206を出力する。上記において、処理2501が本発明に関する部分である。

【0051】図24は、本発明に基づく第三の実施例を表す論理等価検証前処理のフローチャートである。図24は、図1のステップ106～108をステップ2400～2402で置き換えたものである。ステップ2400～2402の処理手順を順次説明する。

【0052】ステップ2400：本ステップは、出力変数単位に、ブール式セットAとブール式セットYのXNOR（Exclusive NOR）論理演算を行い、ブール式セットIを作成する。

【0053】図26は、ブール式セットA500とブール式セットY700をXNOR論理演算したブール式セットIを表す。

【0054】ステップ2401：本ステップは、出力変数単位に、ブール式セットIとブール式セットBの論理和演算を行い、ブール式セットJを作成する。

【0055】図27は、ブール式セットI2600とブール式セットB600を論理和演算したブール式セットJを表す。

【0056】ステップ2402：本ステップは、ブール式セットJ2700を論理等価検証モデル2502として主記憶に出力する。

【0057】図28は、本発明に基づく第四の実施例を表す論理等価検証前処理のフローチャートである。図28は、図10のステップ1001～1003をステップ2400と2800～2801で置き換えたものである。ステップ2400は説明済みであるので、ステップ2800～2801の処理手順を順次説明する。

【0058】ステップ2800：本ステップは、出力変数単位に、ブール式セットIとフィルターセットFを接



続し、ブール式セットKを作成する。

【0059】図29は、ブール式セットI 2600（ブール式2601と2602）とフィルターセットF（ブール式1200と1300）を接続したブール式セットKを表す。

【0060】ステップ2801：本ステップは、ブール式セットK 2900を論理等価検証モデル2502として主記憶に出力する。

【0061】本実施例によれば、ドントケア出力に起因する論理の不確定性を取り除くことが可能である。

【0062】

【発明の効果】本発明によれば、ドントケア出力に起因する論理の不確定性を取り除くことができるので、本発明と従来の論理等価検証方法を併用することにより、ドントケア出力を含む機能レベルの組合せ論理と該組合せ論理から自動生成されたゲート論理の論理等価検証が可能である。

【図面の簡単な説明】

【図1】本発明に基づく第一の実施例を表す論理等価検証前処理のフローチャート。

【図2】本発明の前提となる第一の論理等価検証システム構成を表す図。

【図3】ドントケア出力を含む機能レベルの組合せ論理（真理値表）の例の説明図。

【図4】真理値表300から生成されたゲート論理の例の説明図。

【図5】ブール式セットAの説明図。

【図6】ブール式セットBの説明図。

【図7】ブール式セットYの説明図。

【図8】ブール式セットDの説明図。

【図9】ブール式セットEの説明図。

【図10】本発明に基づく第二の実施例を表す論理等価検証前処理のフローチャート。

【図11】フィルター作成処理のフローチャート。

【図12】出力変数Y1のフィルターの説明図。

【図13】出力変数Y2のフィルターの説明図。

【図14】ステップ1102の処理結果の説明図。

【図15】ブール式セットBのコンプリメントの説明図。

【図16】出力変数Y1のステップ1104と1105（一回目）の処理結果の説明図。

【図17】出力変数Y1のフィルターの一回目の中間結果の説明図。

【図18】出力変数Y1のステップ1104と1105（二回目）の処理結果の説明図。

【図19】出力変数Y1のフィルターの二回目の中間結果の説明図。

【図20】出力変数Y1のステップ1104と1105（三回目）の処理結果の説明図。

【図21】出力変数Y2のステップ1104と1105の処理結果の説明図。

【図22】ブール式セットGの説明図。

【図23】ブール式セットHの説明図。

【図24】本発明に基づく第三の実施例を表す論理等価検証前処理のフローチャート。

【図25】本発明の前提となる第二の論理等価検証システム構成を表す図。

【図26】ブール式セットIの説明図。

【図27】ブール式セットJの説明図。

【図28】本発明に基づく第四の実施例を表す論理等価検証前処理のフローチャート。

【図29】ブール式セットKの説明図。

【図30】積項t0とt1の入力変数Xiに関する「共有」関係の定義の説明図。

【符号の説明】

100…機能論理ファイル、101…ゲート論理ファイル、102～108、1000～1003…論理等価検証前処理プログラム202の処理ステップ、200…処理装置、201…論理生成プログラム、202…論理等価検証前処理プログラム、203、204…論理等価検証前処理後の組合せ論理、205…論理等価検証プログラム、206…検証結果、2400～2402、2800～2801…論理等価検証前処理2501の処理ステップ、2501…論理等価検証前処理、2502…論理等価検証モデル、2503…論理等価検証処理。

【図3】

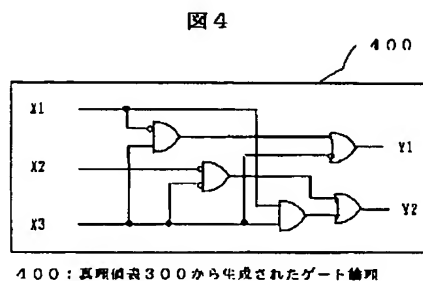
図3

入力変数名	出力変数名	
X1	X2	X3
0	0	0
1	0	0
0	1	0
0	0	1
1	1	0
0	1	1
1	1	1

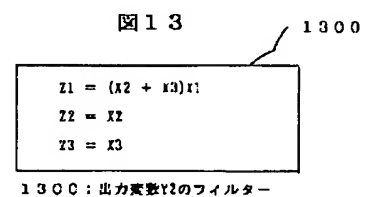
(注) X: ドントケア

300: ドントケア出力を含む機能レベルの組合せ論理 (真理値表)

【図4】



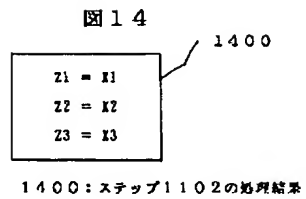
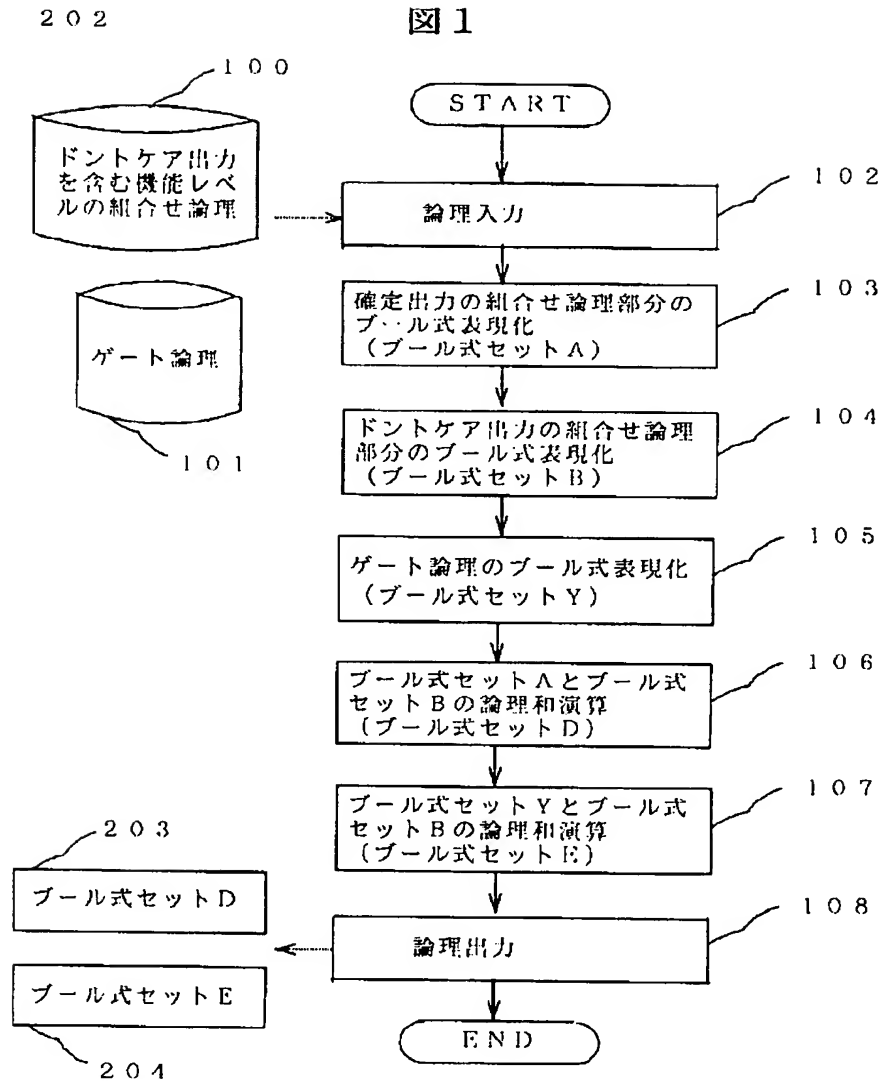
【図13】





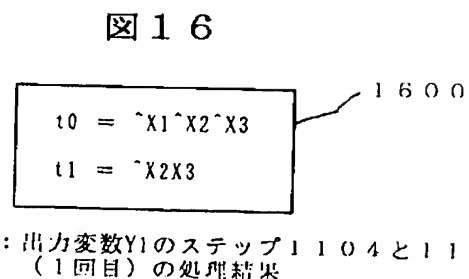
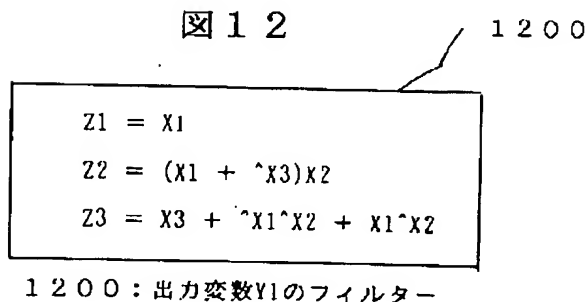
【図1】

【図14】



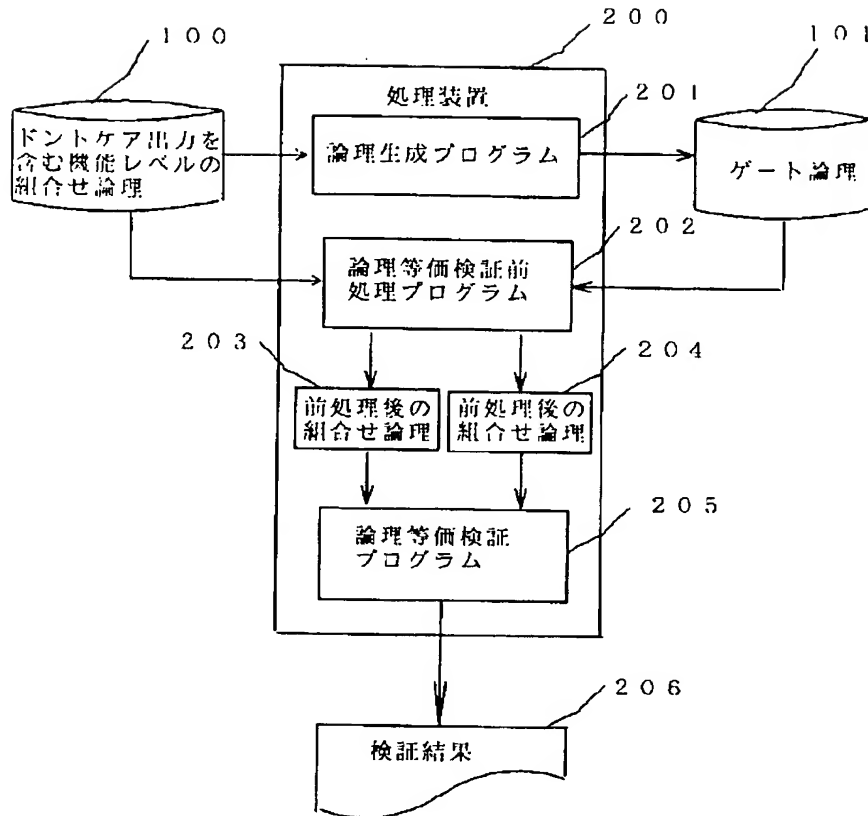
【図12】

【図16】



【図2】

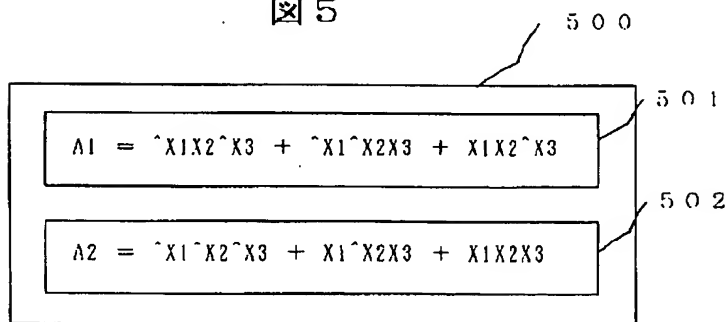
図2



100 : 機能論理ファイル  
101 : ゲート論理ファイル

【図5】

図5



(注)  $\bar{\phantom{x}}$  : not

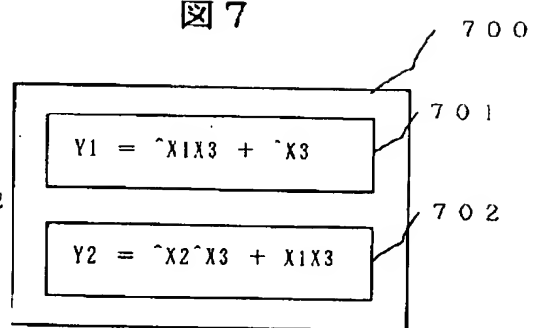
500 : ブール式セットA

501 : Y1の確定出力の組合せ論理部分のブール式

502 : Y2の確定出力の組合せ論理部分のブール式

【図7】

図7



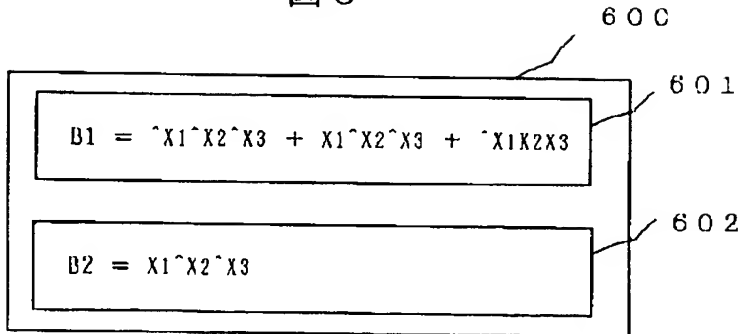
700 : ブール式セットY

701 : Y1のゲート論理部分のブール式

702 : Y2のゲート論理部分のブール式

【図6】

図 6



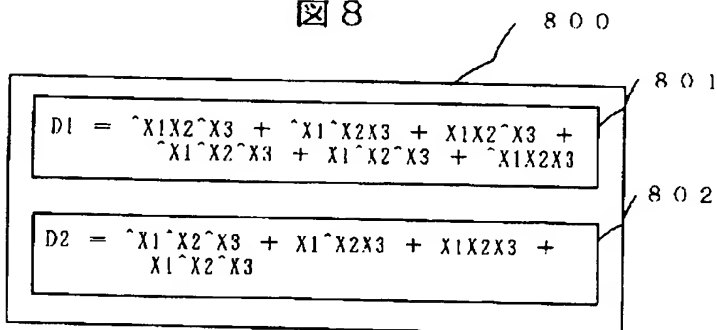
600: ブール式セットB

601: Y1のドントケア出力の組合せ論理部分のブール式

602: Y2のドントケア出力の組合せ論理部分のブール式

【図8】

図 8



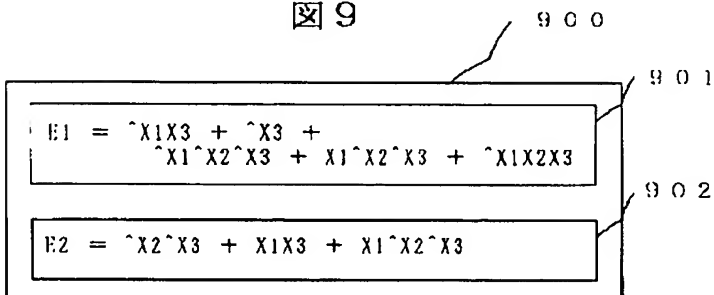
800: ブール式セットD

801: ブール式501と601を論理和演算したブール式

802: ブール式502と602を論理和演算したブール式

【図9】

図 9



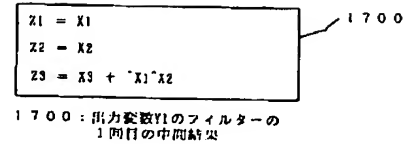
900: ブール式セットE

901: ブール式701と601を論理和演算したブール式

902: ブール式702と602を論理和演算したブール式

【図17】

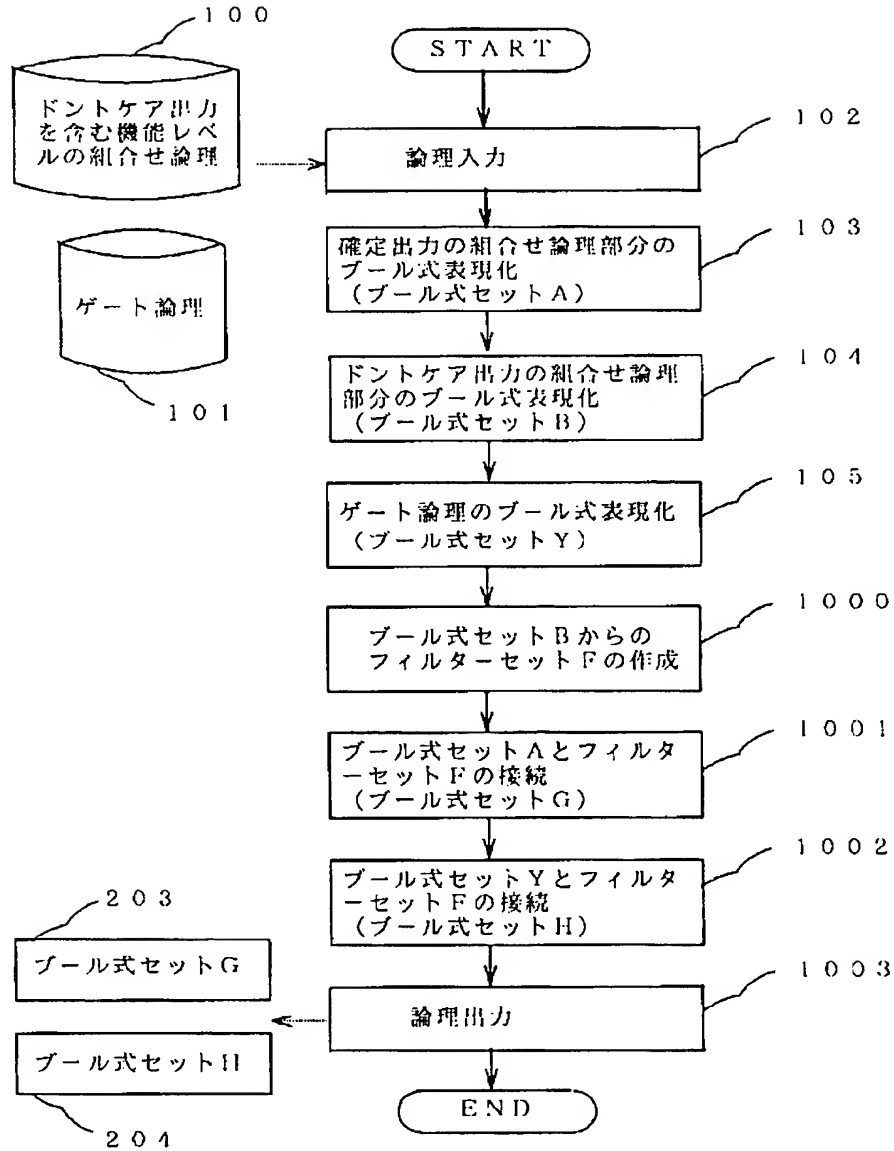
図 17



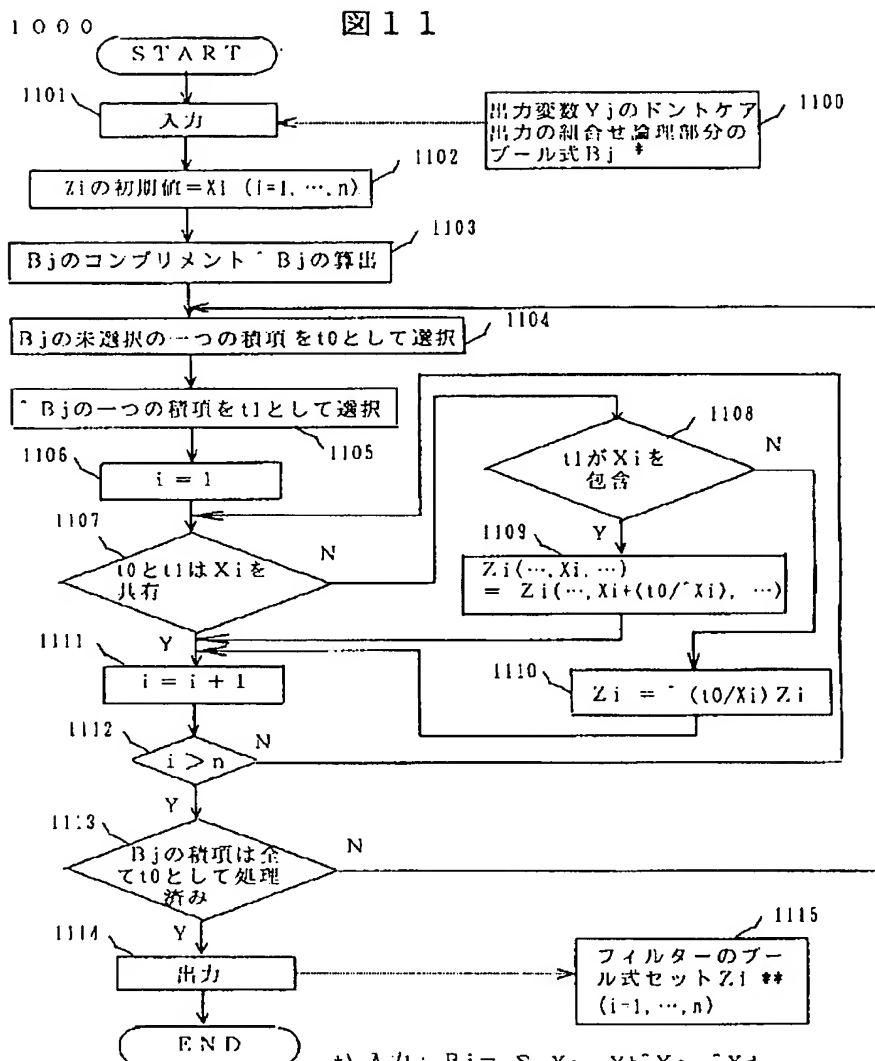
【図10】

202

図10



【図11】



\*) 入力:  $B_j = \sum X_a \cdots X_b \wedge X_c \cdots \wedge X_d$

\*\*) 出力:  $Z_i = Z_i(X_1, \dots, X_n, \wedge X_1, \dots, \wedge X_n)$

ここで、 $X_a, X_b, X_c, X_d, X_1, X_n$ は入力変数  
 $a, b, c, d, i=1, \dots, n$  但し、 $n$  = 入力変数の総数  
 $j=1, \dots, m$  但し、 $m$  = 出力変数の総数

【図15】

図15

$$\text{1501} \quad \text{1502} \quad \bar{B1} = \bar{X2X3} + X2\bar{X3} + X1X2$$

$$\bar{B2} = \bar{X1}^X2 + \bar{X2X3} + X2\bar{X3} + X1X2X3$$

1501: ブール式601のコンプリメント

1502: ブール式602のコンプリメント

【図18】

図18

$$\begin{aligned} t0 &= X1\bar{X2}^X3 \\ t1 &= \bar{X2X3} \end{aligned} \quad \text{1800}$$

1800: 出力変数Y1のステップ1104と1105(2回目)の処理結果

【図19】

図19

$$\begin{aligned} Z1 &= X1 \\ Z2 &= X2 \\ Z3 &= X3 + \bar{X1}^X2 + X1\bar{X2} \end{aligned} \quad \text{1900}$$

1900: 出力変数Y1のフィルターの2回目の中間結果

【図20】

図20

$$\begin{aligned} t0 &= \bar{X1X2X3} \\ t1 &= \bar{X2X3} \end{aligned} \quad \text{2000}$$

2000: 出力変数Y1のステップ1104と1105(3回目)の処理結果

【図21】

図21

$$\begin{aligned} t0 &= X1\bar{X2}^X3 \\ t1 &= \bar{X1}^X2 \end{aligned} \quad \text{2100}$$

2100: 出力変数Y2のステップ1104と1105の処理結果

【図 2 2】

図 2 2

2 2 0 0

$$\begin{aligned}
 G1 &= \bar{Z1}Z2\bar{Z3} + \bar{Z1}\bar{Z2}Z3 + Z1Z2\bar{Z3} \\
 &= \bar{X1}(X1+\bar{X3})X2\bar{(X3+\bar{X1}\bar{X2}+X1\bar{X2})} + \\
 &\quad \bar{X1}\bar{(X1+\bar{X3})X2}(X3+\bar{X1}\bar{X2}+X1\bar{X2}) + \\
 &\quad X1(X1+\bar{X3})X2\bar{(X3+\bar{X1}\bar{X2}+X1\bar{X2})} \\
 G2 &= \bar{Z1}\bar{Z2}\bar{Z3} + Z1\bar{Z2}Z3 + Z1Z2Z3 \\
 &= \bar{(X2+X3)X1}\bar{X2}\bar{X3} + \\
 &\quad (X2+X3)X1\bar{X2}X3 + \\
 &\quad (X2+X3)X1X2X3
 \end{aligned}$$

2 2 0 0 : ブール式セット G

【図 2 3】

図 2 3

2 3 0 0

$$\begin{aligned}
 H1 &= \bar{Z1}Z3 + \bar{Z3} \\
 &= \bar{X1}(X3+\bar{X1}\bar{X2}+X1\bar{X2})\bar{(X3+\bar{X1}\bar{X2}+X1\bar{X2})} + \\
 H2 &= \bar{Z2}\bar{Z3} + Z1Z3 \\
 &= \bar{X2}\bar{X3} + (X2+X3)X1X3
 \end{aligned}$$

2 3 0 0 : ブール式セット H

【図 2 7】

図 2 7

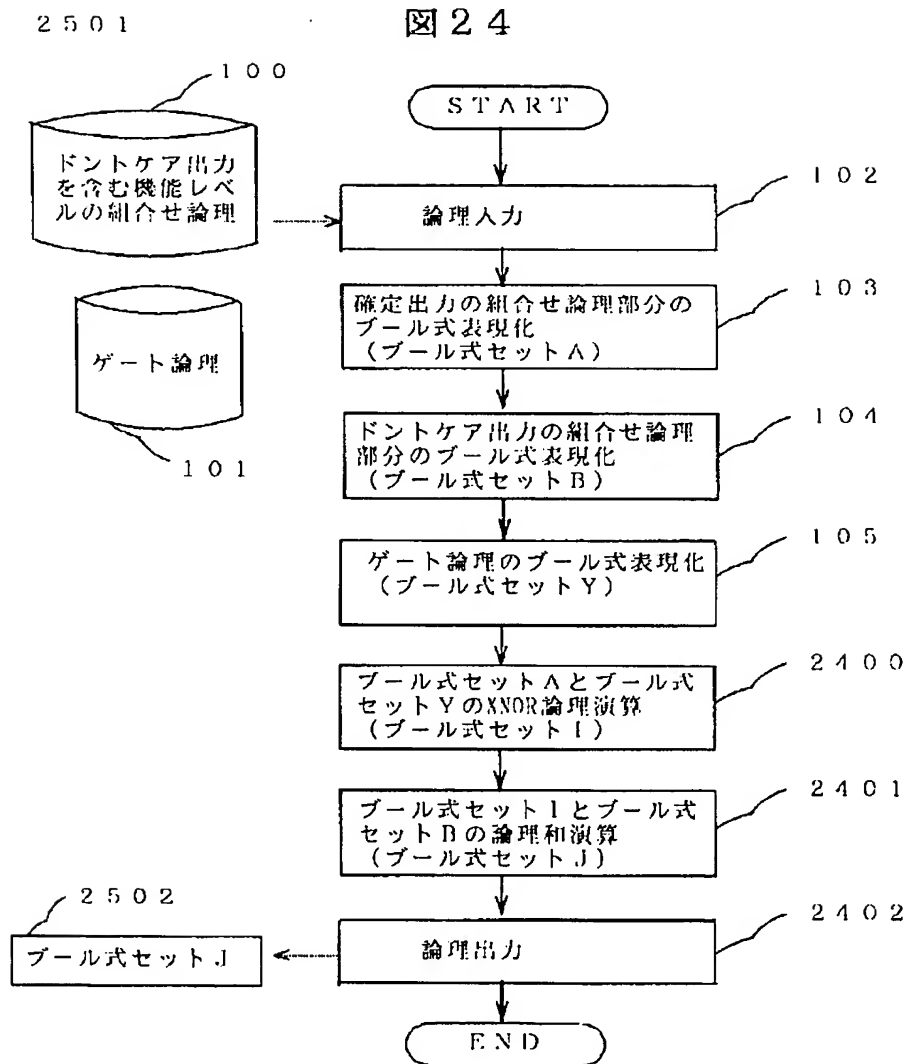
2 7 0 0

$$\begin{aligned}
 J1 &= I1 + B1 \\
 &= (\bar{X1}X2\bar{X3}+\bar{X1}\bar{X2}X3+X1X2\bar{X3})(\bar{X1}X3+\bar{X3}) + \\
 &\quad \bar{(X1X2\bar{X3}+\bar{X1}\bar{X2}X3+X1X2\bar{X3})}(\bar{X1}X3+\bar{X3}) + \\
 &\quad \bar{X1}\bar{X2}\bar{X3} + X1\bar{X2}\bar{X3} + \bar{X1}X2X3 \\
 J2 &= I2 + B2 \\
 &= (\bar{X1}\bar{X2}\bar{X3}+X1\bar{X2}X3+X1X2X3)(\bar{X2}\bar{X3}+X1X3) + \\
 &\quad \bar{(X1\bar{X2}\bar{X3}+X1\bar{X2}X3+X1X2X3)}(\bar{X2}\bar{X3}+X1X3) + \\
 &\quad X1\bar{X2}\bar{X3}
 \end{aligned}$$

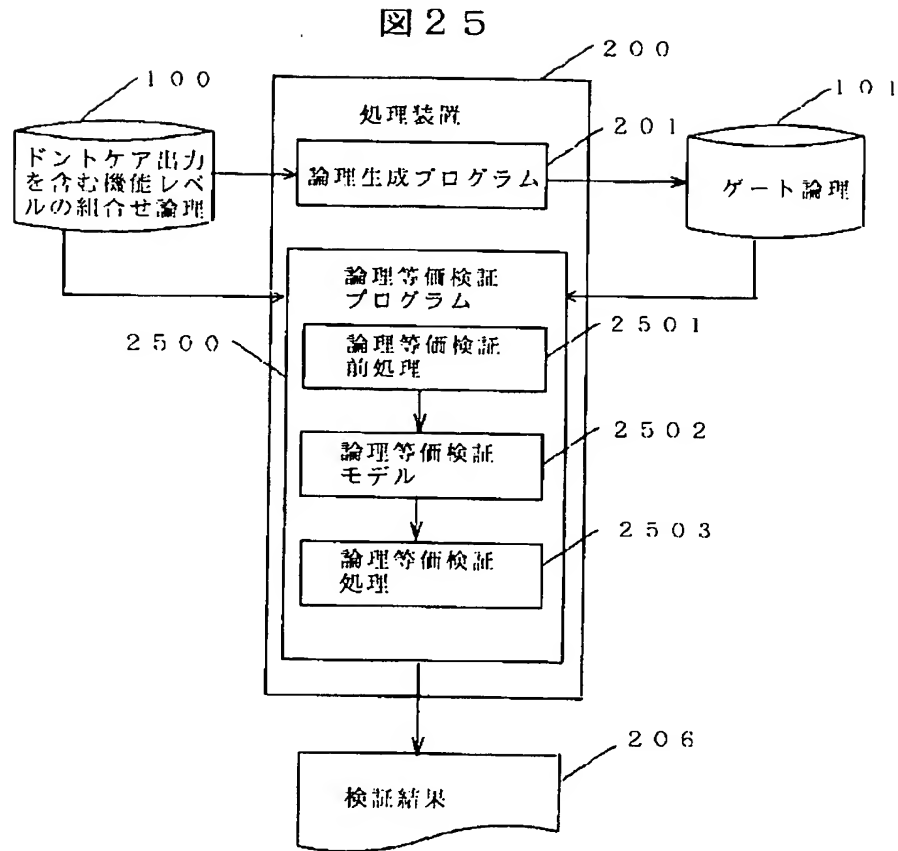
2 7 0 0 : ブール式セット J



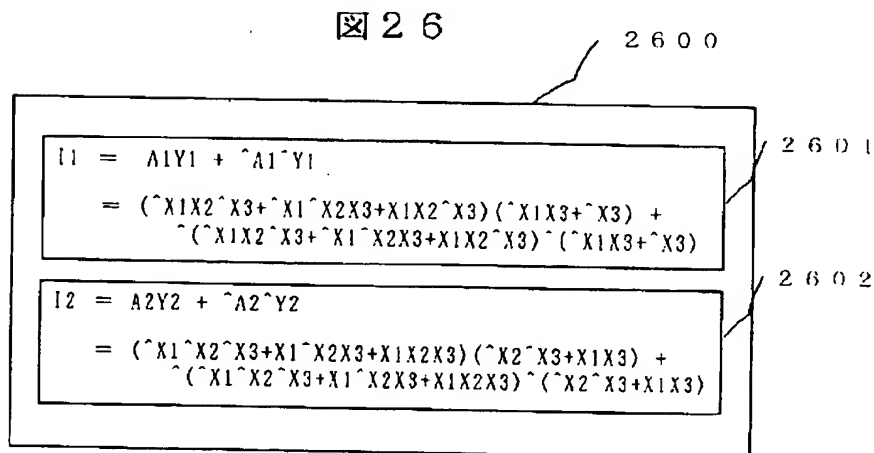
【図24】



【図25】



【図26】



2600: ブール式セット1

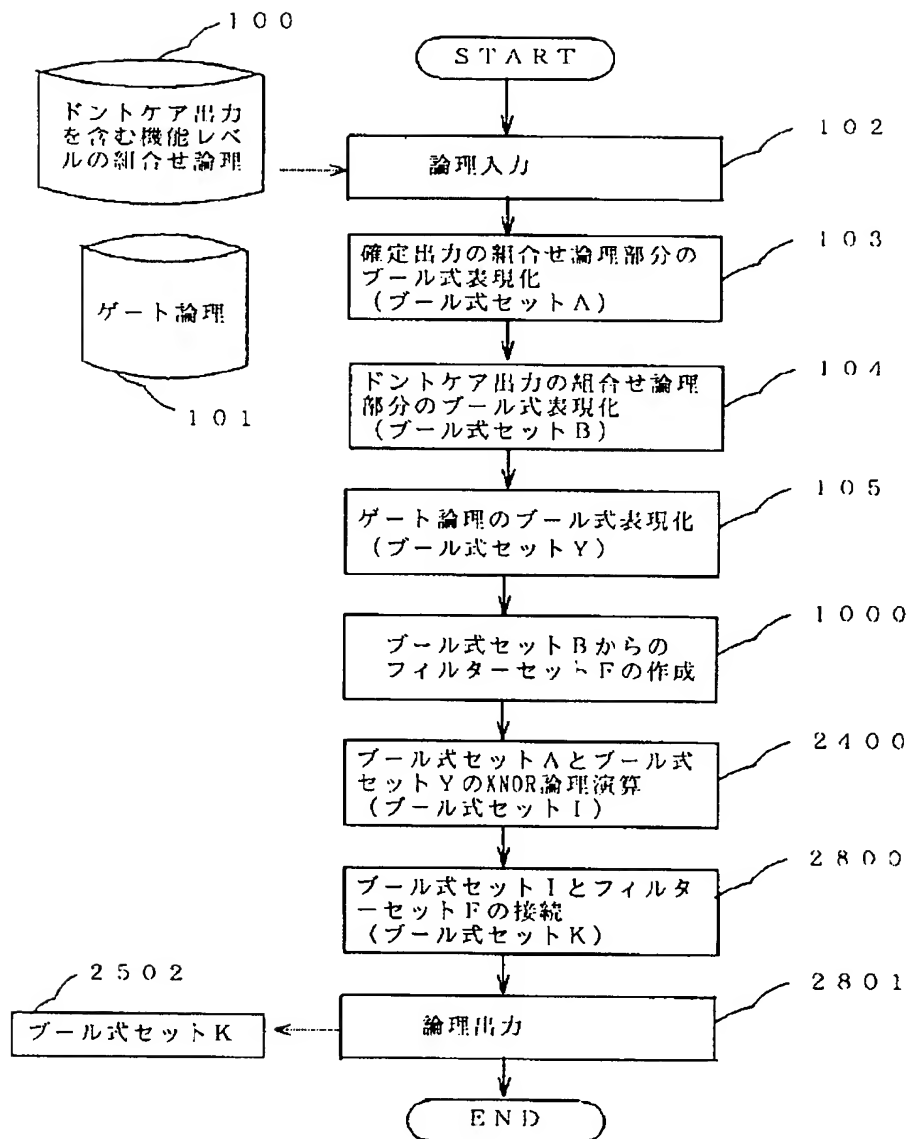
2601: ブール式501と701をXNOR論理演算したブール式

2602: ブール式502と702をXNOR論理演算したブール式

【図28】

2501

図28



【図29】

図29

2900

$$K1 = (^{Z1}Z2^{Z3} + ^{Z1}Z2Z3 + ^{Z1}Z2^{Z3})(^{Z1}Z3 + ^{Z3}) + (^{Z1}Z2^{Z3} + ^{Z1}Z2Z3 + ^{Z1}Z2^{Z3})(^{Z1}Z3 + ^{Z3})$$

$$\begin{aligned} \text{ここで、} \quad Z1 &= X1 \\ Z2 &= (X1 + ^{X3})X2 \\ Z3 &= X3 + ^{X1}X2 + X1^{X2} \end{aligned}$$

$$K2 = (^{Z1}Z2^{Z3} + ^{Z1}Z2Z3 + ^{Z1}Z2Z3)(^{Z2}Z3 + ^{Z1}Z3) + (^{Z1}Z2^{Z3} + ^{Z1}Z2Z3 + ^{Z1}Z2Z3)(^{Z2}Z3 + ^{Z1}Z3)$$

$$\begin{aligned} \text{ここで、} \quad Z1 &= (X2 + X3)X1 \\ Z2 &= X2 \\ Z3 &= X3 \end{aligned}$$

2900: ブール式セットK

【図30】

図30

3000

	t0における Xiの現れ方	t1における Xiの現れ方
t0とt1は 変数Xiを 共有する	$X_i$	$X_i$
	$^{X_i}$	$^{X_i}$
	$X_i$	*
	$^{X_i}$	*
	*	*
t0とt1は 変数Xiを 共有しない	$^{X_i}$	$X_i$
	*	$X_i$
	$X_i$	$^{X_i}$
	*	$^{X_i}$

(注) \* は変数Xiに依存しない場合を表す

ここで、 $i = 1, \dots, n$  ( $n$  = 入力変数の総数)

3000: 積項t0とt1の入力変数Xiに関する「共有」関係の定義

フロントページの続き

(72)発明者 溝上 良人

神奈川県秦野市堀山下1番地 株式会社日

立製作所汎用コンピュータ事業部内